Related-Key Rectangle Cryptanalysis of Rijndael-160 and Rijndael-192

Qingju Wang[1,4⋆], Zhiqiang Liu[1,4⋆], Deniz Toz[1,2⋆], Kerem Varici[1,3⋆], and Dawu Gu[4 ⋆]

[1] ESAT/COSIC, KU Leuven and iMinds
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
[2] Clear2Pay, De Kleetlaan 6A, 1831 Diegem, Brussels, Belgium
[3] ICTEAM-Crypto Group, Universite catholique de Louvain, 1348 Louvain-la-Neuve, Belgium
[4] Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dongchuan Road, Minhang District, Shanghai, 200240, China
qingju.wang@esat.kuleuven.be, ilu_zq@163.com
deniz.toz@gmail.com,kerem.varici@uclouvain.be,dwgu@sjtu.edu.cn

Abstract. In this paper we present the first related-key rectangle cryptanalysis of Rijndael-160/160 and Rijndael-192/192. Our attack on Rijndael-160/160 covers eight rounds. The attack complexities are $2^{126.5}$ chosen plaintexts, $2^{129.28}$ 8-round Rijndael-160/160 encryptions and $2^{132.82}$ bytes. Our attack on Rijndael-192/192 covers ten rounds. It requires 2^{179} chosen plaintexts, $2^{181.09}$ 10-round Rijndael-192/192 encryptions and $2^{185.59}$ bytes memory. These are the currently best cryptanalytic results on Rijndael-160/160 and Rijndael-192/192 in terms of the number of attacked rounds. Furthermore, our results show that the slow diffusion in the key schedule of Rijndael makes it a target for this type of analysis.

Keywords: Rijndael, related-key attack, rectangle cryptanalysis

1 Introduction

Block ciphers are used widely in cryptography to ensure confidentiality and authenticity. They are needed both in software and in hardware. For example, they are used in electronic payments or for wireless security. For different demands, different algorithms are designed [1–3] and they have been standardised as well. Apart from this main functionality, block ciphers are also used as underlying primitives in the design of hash functions or pseudo-random number generators.

Rijndael [2] is a block cipher designed by Daemen and Rijmen and is a substitution-permutation network following the wide-trail strategy. Both the block length and the key length can be any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits, independently of each other with key size greater than or equal to block size. The 128-bit block variant of Rijndael has been chosen as the Advanced Encryption Standard (AES) [4]. This paper deals with non-AES Rijndael variants, that is, Rijndael-b/k where b indicates the block size and k indicates the key size in bits.

Without doubt AES is one of the most well-studied block ciphers: since its introduction 15 years ago there has been extensive analysis of AES. Some prominent examples include square attacks, impossible differential attacks, boomerang attacks, rectangle attacks and meet-in-the-middle attacks in both the single-key and related-key settings [5–18].

On the other hand, the variants of Rijndael with larger block sizes have got arguably less attention from the cryptographic community. Current analysis includes several multiset and integral attacks [19–22], as well as impossible differential cryptanalysis [23–25]. A summary of these attacks and their time and data complexities are given in Table 1. An important motivation for the study of large-block Rijndael is the deployment of Rijndael-like permutations in the design of hash functions, Whirlwind [26], SHAvite-3 [27], Whirlpool [28], ECHO [29], PHOTON [30] and SHA-3 finalist Grøstl [31] constituting some especially interesting instances.

The rectangle attack [32] introduced by Biham et al. is a special type of differential cryptanalysis. The main idea of the attack is to divide the cipher E into two sub-ciphers E_0 and E_1 such that $E = E_1 \circ E_0$. The

⋆ Corresponding authors.

attacker then constructs two relatively short differentials for E_0 and E_1 instead of finding a long differential for the block cipher E. After that, a rectangle distinguisher for E can be established by combining these two short differentials delicately. This technique is useful when we have short differentials with high probability instead of long ones with lower differential probabilities.

In the related-key model, the attacker can decrypt/encrypt not only under the master key K, but also under the keys $f_1(K), f_2(K) \dots f_m(K)$, which are called related-keys. The relations f_i are chosen by the attacker in advance. The aim of the attacker is to recover the master keys. The first related-key attacks consider simple mappings, for example, rotations [33] and bit flips [34]. Recent attack on AES [18] exploits the difference not between the master keys but between the subkeys. The extra control might make the attack harder to mount in practice. However, the designers still usually make great efforts to build ideal primitives which can be used without further cryptanalysis to the applications of modes of operation or protocols. Related-key attacks are applied to the attacks on the protocols that use ciphers as a building block [35]. Contrasting to the single-key attack, related-key attack recovers a secret parameter of the protocol. Therefore resistance to related-key attack becomes one of the important design aims for block ciphers, actually this was also stated as one of the design goals of the Rijndael.

Moreover it is also possible to combine rectangle-type attack with related-key attack to derive a more efficient cryptanalytic approach [36]. Actually, this type of combined approach has been applied to various block ciphers and some intriguing results have been achieved for AES [18, 37] and KASUMI [38, 39].

Contributions. In this paper we propose related-key rectangle attacks on reduced-round versions of Rijndael-160/160 and Rijndael-192/192. Our attacks use the idea of switch technique [40] and local collisions in [18]. We construct 6 and 8-round rectangle distinguishers of Rijndael-160/160 and Rijndael-192/192, based on which we attack 8 and 10 rounds of these two ciphers respectively. To our knowledge, our results are the best ones in terms of the number of attacked rounds. The attacks on Rijndael-160/160 and Rijndael-192/192 are summarized in Table 1.

Table 1. Summary of Attacks on Rijndael-160/160 and Rijndael-192/192

Cipher	# Rounds	Complexity Time(EN)	Complexity Data(CP)	Memory(Bytes)	Attack Type	Ref.
Rijndael-160/160	6	2^{135}	$2^{105.5}$	-	Imp. Diff.	[23]
	6	$2^{81.9}$	2^{147}	-	Imp. Diff.	[24]
	7	2^{144}	$2^{130.6}$	2^{128}	Multiset	[20]
	7	$2^{81.9}$	2^{147}	-	Imp. Diff.	[24]
	7	2^{108}	$2^{94.6}$	2^{92}	Integral	[41]
	7	$2^{98.6}$	$2^{98.6}$	-	Integral	[22]
	8	$2^{129.28}$	$2^{126.5\dagger}$	$2^{132.82}$	**RK Rectangle**	**§5**
Rijndael-192/192	6	2^{151}	$2^{121.5}$	2^{93}	Imp. Diff.	[23]
	6	$2^{113.8}$	$2^{93.2}$	2^{93}	Imp. Diff.	[24]
	7	2^{120}	$2^{128} - 2^{119}$	2^{61}	Partial Sum	[19]
	7	2^{144}	$2^{130.6}$	2^{128}	Multiset	[20]
	7	$2^{66.6}$	$2^{66.6}$	-	Integral	[22]
	7	$2^{174.5}$	$2^{28.5}$	-	Integral	[22]
	8	2^{188}	$2^{128} - 2^{119}$		Partial Sum	[19]
	8	$2^{177.4}$	2^{158}	2^{157}	Imp. Diff.	[24]
	8	$2^{81.4}$	2^{179}	2^{61}	Imp. Diff.	[24]
	8	$2^{174.5}$	$2^{68.5}$	-	Integral	[22]
	8	$2^{162.6}$	$2^{162.6}$	-	Integral	[22]
	9	$2^{174.5}$	$2^{164.5}$	-	Integral	[22]
	10	$2^{181.09}$	$2^{179\ddagger}$	$2^{185.59}$	**RK Rectangle**	**§6**

CP: Chosen Plaintext; EN: Encryptions
† $2^{128.5}$ ciphertexts under 4 related keys; ‡ 2^{181} ciphertexts under 4 related keys

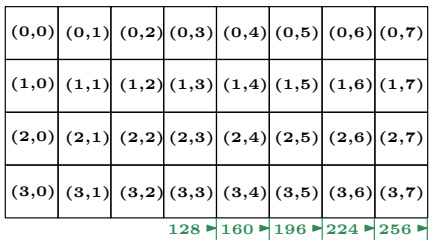
Outline. This paper is organized as follows. In Section 2 we give a brief description of Rijndael and the notations used in our analysis. Section 3 introduces the rectangle attack briefly and shows our rectangle distinguishers. We demonstrate our attacks on Rijndael-160/160 and Rijndael-192/192 in Sections 5 and 6, respectively. Finally, Section 7 concludes this paper.

2 Description of Rijndael and Notations

In Rijndael, each data block (plaintext, ciphertext, subkey, or intermediate step) is represented by a $4 \times N_b$ `state matrix` of bytes, where N_b is the block size divided by 32. The state is then transformed by iterating a round function. The round function is composed of the following four operations:

- `SubBytes (SB)` : a non-linear byte substitution (8×8-bit S-box) that acts on every byte of the state.
- `ShiftRows (SR)`: a cyclic shift of bytes in a row that acts individually on each of the last three rows of the state. The shift offset C_i of row i depends on the block length N_b (See Fig. 1).
- `MixColumns (MC)`: a linear transformation (based on an $[8, 4, 5]$ MDS code over $GF(2^8)$) that acts independently on every column of the state
- `AddRoundKey (AK)`: the exclusive-or of the round key with the intermediate state.

The number of rounds for the cipher N_r varies with N_b and N_k (the key size divided by 32). Before the first round, there exists a whitening layer consisting of `AddRoundKey` only, and in the last round the `MixColumns` operation is omitted. We assume that this is also the case for the reduced round versions of Rijndael.



(a) Byte index

N_k \ N_b	4	5	6	7	8
4	10	11	12	13	14
5	11	11	12	13	14
6	12	12	12	13	14
7	13	13	13	13	14
8	14	14	14	14	14

(b) Number of rounds

N_b	C_0	C_1	C_2	C_3
4	0	1	2	3
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

(c) Shift offsets

Fig. 1. Byte Index of the State Matrix and the Shift Offsets for Each Block Length N_b

Key Scheduling: The key schedule derives $(N_r + 1)$ b-bit round keys RK_0, $RK_1 \ldots RK_{N_r}$ from the master key. It consists of a linear array of 4-byte words denoted by $W[i]$ for $0 \leq i \leq N_b \cdot (N_r + 1)$. The first N_k words $W[0] \| W[1] \| \cdots \| W[N_k - 1]$ are directly initialized with the words of the master key, while the remaining key words, $W[i]$ for $i \in [N_k, N_k \cdot (N_r + 1) - 1]$ are generated by the following algorithm:

`if` $(i \bmod N_k = 0)$ `then` $W[i] = W[i - N_k] \oplus SB(W[i - 1] \lll 8) \oplus Rcon[i/N_k]$
`else if` $(N_k > 6$ and $i \bmod N_k = 4)$ `then` $W[i] = W[i - N_k] \oplus SB(W[i - 1])$
`else` $W[i] = W[i - N_k] \oplus W[i - 1]$

where $\lll$ denotes the rotation of the word to the left and $Rcon[\cdot]$ denotes the fixed constants. Then the round key RK_i is given by the words $W[N_b \cdot i]$ to $W[N_b \cdot (i + 1)]$. The key expansion and the the round key selection of Rijndael-160/160 are illustrated in Figure 2. Here we only give a brief description of Rijndael, for more detailed specification of the cipher, we refer to [2].

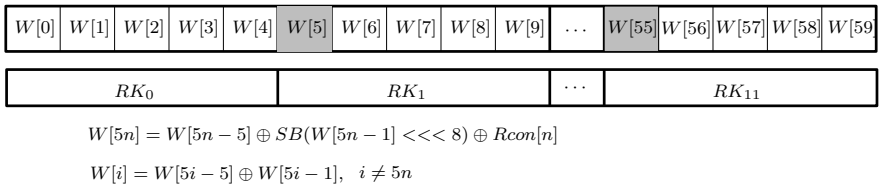


Fig. 2. Key expansion and round key selection for $N_b = N_k = 5$.

Notation: The notation that we will use throughout this paper is as follows:

P_a, P_b, P_c, P_d	the plaintexts
$(P_a)_{i,j}$,$(P_b)_{i,j}$,$(P_c)_{i,j}$,$(P_d)_{i,j}$	the byte at row i column j of the plaintext state
C_a, C_b, C_c, C_d	the ciphertexts
K_a, K_b, K_c, K_d	secret related keys
$(K_a)_{i,j}$, $(K_b)_{i,j}$, $(K_c)_{i,j}$, $(K_d)_{i,j}$	the byte at row i column j of the secret related keys
K_a^r, K_b^r, K_c^r, K_d^r	secret subkey of K_a, K_b, K_c and K_d in round r
ΔK_{ab}^r, ΔK_{ac}^r, ΔK_{cd}^r, ΔK_{bd}^r	the difference of the related keys in round r
$(\Delta K_{ab}^r)_{i,j}$	difference byte of state ΔK_{ab}^r, at position row i and column j
$\mathtt{SB}[(i,j)]$	the SB operation on the byte at row i column j of the state matrix
$\mathtt{SB}[\{(i,j)\}]$	the SB operation on the subset bytes $\{(i,j)\}$ of the state matrix
$E(\cdot,\cdot)$	encryption operation defined as $\{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$

3 Rectangle Attack

The rectangle attack introduced by Biham et al. [32] aims to reduce the complexity of the differential cryptanalysis. The main idea is to use two short differential characteristics with high probabilities instead of one long characteristic with a lower probability. It is also possible to combine rectangle attack with related-key attack to derive the related-key rectangle attack [36] in which the attacker can query to the cipher with other keys that have a specified relation (often an xor-difference) with the original key.

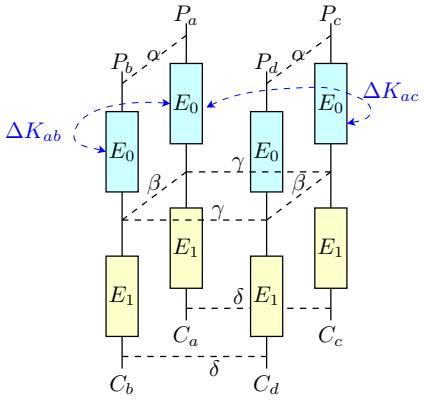


Fig. 3. The Related-Key Rectangle Distinguisher

Let the encryption function E of the block cipher be considered as a cascade of two sub-ciphers $E = E_1 \circ E_0$. Assume that there exists a related-key differential $\alpha \rightarrow \beta$ for E_0 under the key difference ΔK_{ab} with probability p, i.e., $(\Pr[E_0(P,K) \oplus E_0((P \oplus \alpha),(K \oplus \Delta K_{ab})) = \beta] = p)$. Similarly, assume that there exists a related-key differential $\gamma \rightarrow \delta$ for E_1 under the key difference ΔK_{ac} with probability q, where ΔK_{ab}

and ΔK_{ac} are the key differences known by the attackers. A related-key rectangle distinguisher is then as follows:

1. Choose N plaintext pairs (P_a, P_b) with $P_b = P_a \oplus \alpha$ at random. Ask for the encryption of P_a under K_a and of P_b under K_b, respectively, where $K_b = K_a \oplus \Delta K_{ab}$.
2. Choose N plaintext pairs (P_c, P_d) with $P_d = P_c \oplus \alpha$ at random. Ask for the encryption of P_c under K_c and of P_d under K_d, respectively, where $K_c = K_a \oplus \Delta K_{ac}$ and $K_d = K_c \oplus \Delta K_{ab} = K_b \oplus \Delta K_{ac}$.
3. For a quartet of plaintexts (P_a, P_b, P_c, P_d) with corresponding ciphertexts (C_a, C_b, C_c, C_d), check whether $C_a \oplus C_c = C_b \oplus C_d = \delta$ holds or not. If yes, we call it a *right rectangle quartet*. Fig. 3 illustrates the related-key rectangle distinguisher.

The related-key rectangle attack can be mounted for all possible β's and γ's simultaneously. Firstly, we can use any γ for which $\gamma \xrightarrow{E_1} \delta$ holds. This is equivalent to mounting the attack for all values of γ with the condition $(E_0(P_a), E_0(P_c))$ and $(E_0(P_b), E_0(P_d))$ have the same difference (γ). In this case the probability that conditions of the distinguisher are satisfied is

$$2^{-n} p^2 \sum_{\gamma \xrightarrow{E_1} \delta} Pr^2[\gamma \xrightarrow{E_1} \delta] = 2^{-n} p^2 \hat{q}^2,$$

where n is the block size, and $\hat{q} = \sqrt{\sum_{\gamma \xrightarrow{E_1} \delta} Pr^2[\gamma \xrightarrow{E_1} \delta]}$. Similarly, we can use all β values simultaneously as well. Conditions for this case become: $E_0(P_a) \oplus E_0(P_b) = E_0(P_c) \oplus E_0(P_d) = \beta$ and $E_0(P_a) \oplus E_0(P_c) = \gamma$ and the quartet has probability $Pr^2[\gamma \to \delta]$ to become a right quartet. Hence, the probability that a given quartet is a right quartet is

$$2^{-n} \sum_{\alpha \xrightarrow{E_0} \beta} Pr^2[\alpha \xrightarrow{E_0} \beta] \sum_{\gamma \xrightarrow{E_1} \delta} Pr^2[\gamma \xrightarrow{E_1} \delta] = 2^{-n} \hat{p}^2 \hat{q}^2,$$

where $\hat{p} = \sqrt{\sum_{\alpha \xrightarrow{E_0} \beta} Pr^2[\alpha \xrightarrow{E_0} \beta]}$. Therefore starting with N plaintext pairs with difference α, we expect to find $N^2 2^{-n} (\hat{p}\hat{q})^2$ right quartets. For an ideal cipher, Step 3 is expected to hold with probability 2^{-2n}. Therefore, if $\hat{p}\hat{q} \gg 2^{-n/2}$, the algorithm above allows to distinguish E from an ideal cipher. We refer to [32, 36, 42, 43] for more detail.

3.1 Local Collision

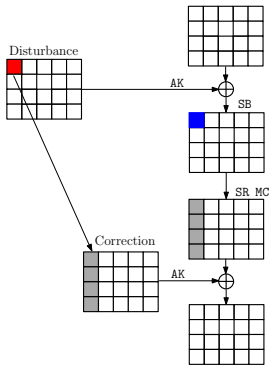


Fig. 4. A Local Collision of Rijndael-160/160

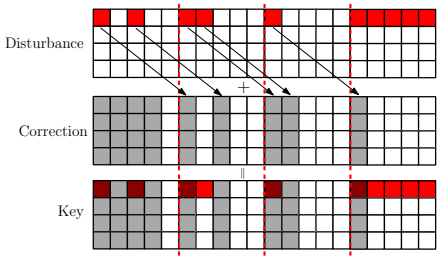


Fig. 5. Constructing Related-keys from Local Collisions

A local collision is a differential that starts and ends with the zero difference in the internal state, but is non-zero in the middle. The idea of local collisions has been first introduced by Joux and Chabaud [44] to

attack hash functions. It aims to inject a difference (called *disturbance*, in red) into an intermediate step and then to *correct* the resulting differences with the injections in the next steps (called *correction*, in grey) to obtain a collision. The goal is to reduce the complexity of the attack by having as few disturbances as possible. This idea has been later successfully applied to block ciphers in [18]. A local collision of Rijndael-160/160 is shown in Fig. 4.

In order to construct an optimal trail, first of all we construct a minimal-weight disturbance layer, which will become a part of the key schedule difference. Then, correction layer is constructed by encrypting on the previous round of the disturbance layer. The key schedule difference is the sum of the disturbance and the correction layers. The 4-round Rijndael-160/160 key schedule difference constructed from local collisions is illustrated in Fig. 5. We follow this approach in our analysis of Rijndael-160/160 and Rijndael-192/192.

3.2 The Related Keys

In order to mount the related-key attacks presented in this paper, the adversary needs to construct the relations between different keys as follows.

Regarding Rijndael-160/160, for a secret key K_a, which the attacker tries to find, we define a simple form of this relation as xor with a constant to obtain K_b: $K_b = K_a \oplus \Delta K^0_{ab}$, where the constant ΔK^0_{ab} is chosen in advance (see Table 2). Then we compute the subkeys K^4_a and K^4_b, based on which, the subkeys K^4_c and K^4_d can be calculated by using the subkey difference ΔK^4_{ac}. After that, according to the key schedule of Rijndael-160/160, we can derive K_c and K_d from the subkeys K^4_c and K^4_d, respectively. This is depicted in Fig. 6.

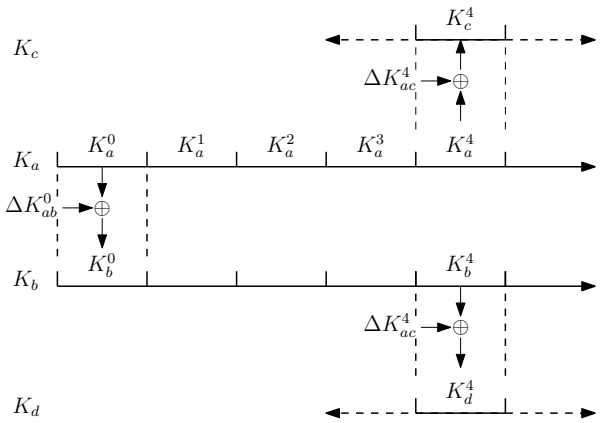


Fig. 6. The Related-Key Computation of Rijndael-160/160

For Rijndael-192/192, a more complex non-linear forms of the relation between the keys will be adopted. We choose a desired XOR relation in the second subkey as ΔK^1_{ab}, and then define the implied relation between the actual keys K_a and K_b as: $K_b = F^{-1}(F(K_a) \oplus \Delta K^1_{ab})$ where F represents a single round of the Rijndael-192/192 key schedule. Similar to Rijndael-160/160, we can define the relation between K_b, K_c and K_d for Rijndael-192/192 (see Table 3).

3.3 Rectangle Switch

In this Subsection we focus on the transition between the top characteristic E_0 and the bottom characteristic E_1 of the rectangle. This method is called *switch technique* [40] and it has been used to improve the probability of boomerang distinguisher [18, 39, 45]. In this paper, we apply this switch technique to our rectangle distinguishers on Rijndael. Let E be $(m + n)$-round Rijndael cipher. Then, the common application is to choose

$$E_0 = (AK \circ MC \circ SR \circ SB)^m$$
$$E_1 = (AK \circ MC \circ SR \circ SB)^n$$

Table 2. Related-key Differences for Rijndael-160

ΔK^r_{ab}															
0	3f	3e	3f	3e	00	1	3f 01 3e 00 00		2	3f 3e 00 00 00		3	3f 01 01 01 01		
	1f	1f	1f	1f	00		1f 00 1f 00 00			1f 1f 00 00 00			1f 00 00 00 00		
	1f	1f	1f	1f	00		1f 00 1f 00 00			1f 1f 00 00 00			1f 00 00 00 00		
	21	21	21	21	00		21 00 21 00 00			21 21 00 00 00			21 00 00 00 00		
ΔK^r_{ac}															
4	21	21	21	21	00	5	21 00 21 00 00		6	21 21 00 00 00		7	21 00 00 00 00		
	3e	3f	3e	3f	00		3e 01 3f 00 00			3e 3f 00 00 00			3e 01 01 01 01		
	1f	1f	1f	1f	00		1f 00 1f 00 00			1f 1f 00 00 00			1f 00 00 00 00		
	1f	1f	1f	1f	00		1f 00 1f 00 00			1f 1f 00 00 00			1f 00 00 00 00		
8	$21 \oplus x^*$	$21 \oplus x$	$21 \oplus x$	$21 \oplus x$	$21 \oplus x$										
	3e	3f	3e	3f	3e										
	1f	1f	1f	1f	1f										
	1f	1f	1f	1f	1f										

* x might differ for ΔK^8_{ac} and ΔK^8_{bd}

Table 3. Related-key Differences for Rijndael-192

ΔK^r_{ab}																	
0	? 3e 00 00 3f 3e					1	3f	01	01	01	3e	00	2	3f 3e 3f 3e 00 00			
	? 1f 00 00 1f 1f						1f	00	00	00	1f	00		1f 1f 1f 1f 00 00			
	? 1f 00 00 1f 1f						1f	00	00	00	1f	00		1f 1f 1f 1f 00 00			
	? 21 00 00 21 21						21	00	00	00	21	00		21 21 21 21 00 00			
3	3f 01 3e 00 00 00					4	3f	3e	00	00	00	00	5	3f 01 01 01 01 01			
	1f 00 1f 00 00 00						1f	1f	00	00	00	00		1f 00 00 00 00 00			
	1f 00 1f 00 00 00						1f	1f	00	00	00	00		1f 00 00 00 00 00			
	21 00 21 00 00 00						21	21	00	00	00	00		21 00 00 00 00 00			
ΔK^r_{ac}																	
6	00 21 21 21 21 00					7	00	21	00	21	00	00	8	00 21 21 00 00 00			
	00 3e 3f 3e 3f 00						00	3e	01	3f	00	00		00 3e 3f 00 00 00			
	00 1f 1f 1f 1f 00						00	1f	00	1f	00	00		00 1f 1f 00 00 00			
	00 1f 1f 1f 1f 00						00	1f	00	1f	00	00		00 1f 1f 00 00 00			
9	00 21 00 00 00 00					10	x^*	$21 \oplus x$	$21 \oplus x$	$21 \oplus x$	$21 \oplus x$	$21 \oplus x$					
	00 3e 01 01 01 01						00	3e	3f	3e	3f	3e					
	00 1f 00 00 00 00						00	1f	1f	1f	1f	1f					
	00 1f 00 00 00 00						00	1f	1f	1f	1f	1f					

* x might differ for ΔK^{10}_{ac} and ΔK^{10}_{bd}

However, due to the flexibility of the SB operation (i.e., it is applied to each byte in the state independently), this choice of E_0, E_1 can be done in a more clever way. Let $\{(i,j)\}$ and $\{(i',j')\}$ be subsets of the state set, SB$[\{(i,j)\}]$ and SB$[\{(i',j')\}]$ be the SB operations on the bytes $\{(i,j)\}$ and $\{(i',j')\}$, respectively. Then E_0 and E_1 can alternatively be defined as follows:

$$\begin{aligned} E_0 &= SB[\{(i,j)\}] \circ (AK \circ MC \circ SR \circ SB)^m, \\ E_1 &= (AK \circ MC \circ SR \circ SB)^{n-1} \circ AK \circ MC \circ SR \circ SB[\{(i',j')\}], \end{aligned} \tag{1}$$

where $\{(i,j)\}$ is the absolute complement of $\{(i',j')\}$ in the state set, and the bytes (i,j) and (i',j') are passive in E_0 and E_1, respectively. For example, in our rectangle distinguisher of Rijndael-160/160, we take $m = 2, n = 4$. The first subcipher E_0 covers rounds 2–3 of Rijndael-160/160 and SB operations on 12 bytes (i,j) in round 4, where $1 \le i \le 3$ and $1 \le j \le 4$; The second subcipher E_1 starts with the SB operations on 8 bytes $(i',0)$, $(0,j')$ in round 4 ($1 \le i' \le 3$, $0 \le j' \le 4$), followed by $AK \circ MC \circ SR$ and rounds 5–7. Our 6-round rectangle distinguisher is illustrated in Fig. 7.

4 The Rectangle Distinguishers

In the analysis of Rijndael-160/160, we use a 6-round rectangle distinguisher, and extend one round before and after the distinguisher respectively (see Fig. 7). Our rectangle distinguisher covers rounds 2–7 and we use the switch technique in round 4 to avoid the active S-boxes in the key schedule and hence to reduce the complexity of our attack. We take $m = 2$ and $n = 4$ in Equation (1) to obtain E_0 and E_1.

The plaintext difference α is specified in 16 bytes, two of them (denoted in dark green) can take any value whereas the remaining ones (denoted in gray) are fixed to $\texttt{(0x3e,0x1f,0x1f,0x21)}^{\texttt{T}}$. The key difference is chosen such that when it is xored to the state, all differences cancel each other except the two bytes at $(0, 0)$ and $(0, 2)$ of the top characteristic. For the differences in these two active bytes we have:

$$(\texttt{0x01} \oplus \alpha_{0,i}) \xrightarrow{SB} \texttt{0x1f}, \qquad i \in \{0, 2\} \tag{2}$$

This guarantees that the input differences to the S-box operations in all the internal states (except the ones specified in Equation 2) are `0x01`. For the active bytes in round 2 of the top characteristic, we adopt `0x1f` as the output difference of `SB` operation in order to achieve the optimal differential probability 2^{-6}. We develop the bottom characteristic by taking an similar approach. As to the active byte in round 3 of the top characteristic, there are 127 possibilities of the output difference for the input difference 0x01 according to the differential distribution table of `SB` operation, among which one happens with the probability 2^{-6}, the others happen with probability 2^{-7}. Then we construct the 6-round related-key rectangle distinguisher by combining the 127 top characteristics and one bottom characteristic mentioned above.

The probability of the 6-round distinguisher can be computed as follows.

- There are three active S-boxes in rounds 2–3, thus the probability of the 127 differentials for E_0 can be calculated as $\hat{p} = \sqrt{(2^{-6})^{2\cdot 2}[1 \cdot (2^{-6})^2 + 126 \cdot (2^{-7})^2]} \approx 2^{-10.5}$.
- There are five active S-boxes in rounds 4–7, thus the probability of the differential for E_1 is $\hat{q} = (2^{-6})^5 = 2^{-30}$.
- In total, the probability of this distinguisher can be calculated as $(2^{-10.5} \cdot 2^{-30})^2 \cdot 2^{-160} = 2^{-251}$.

Similarly, we find a 8-round rectangle distinguisher for Rijndael-192/192 which covers rounds 2–9, and the rectangle switch technique is applied at round 6 (see Fig. 8). There are 9 active S-boxes in the differential characteristics of E_0 (Note that for the active S-box in round 5, all the 127 possible output differences are used to derive 127 characteristics), and the probability can be computed as $\hat{p} = \sqrt{(2^{-6})^{2\cdot 8}[1 \cdot (2^{-6})^2 + 126 \cdot (2^{-7})^2]} \approx 2^{-51.5}$. For the differential characteristic of E_1, there are 5 active S-boxes and the probability is $\hat{q} = (2^{-6})^5 = 2^{-30}$. In total, the distinguisher holds with probability $(2^{-51.5} \cdot 2^{-30})^2 \cdot 2^{-192} = 2^{-355}$.

Moreover, the differences after the `MC` operations (denoted in gray) are given as:

$$\begin{pmatrix} \texttt{0x1f} \\ 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} \texttt{0x3e} \\ \texttt{0x1f} \\ \texttt{0x1f} \\ \texttt{0x21} \end{pmatrix}; \quad \begin{pmatrix} 0 \\ \texttt{0x1f} \\ 0 \\ 0 \end{pmatrix} \xrightarrow{MC} \begin{pmatrix} \texttt{0x21} \\ \texttt{0x3e} \\ \texttt{0x1f} \\ \texttt{0x1f} \end{pmatrix}$$

5 Attack on 8-Round Rijndael-160/160

By using the 6-round distinguisher for round 2–7 given in Subsection 4, we now present a key recovery attack on 8-round Rijndael-160/160 (round 1–8). Based on Fig. 7, an adversary aims to collect sufficient plaintext quartets such that among these plaintext quartets there are averagely 4 right quartets with respect to the 6-round distinguisher. Then among all the collected plaintext quartets, the expected number of quartets satisfying the input and output differences of this distinguisher for a random permutation is $(4/2^{-251}) \cdot 2^{-320} = 2^{-67}$. From this the adversary could distinguish the correct value from the wrong guessed values of the subkey bytes adopted in rounds 1 and 8 which are related to the 6-round distinguisher with a high probability. The attack procedure is divided into two phases: *data collection* phase and *key recovery* phase.

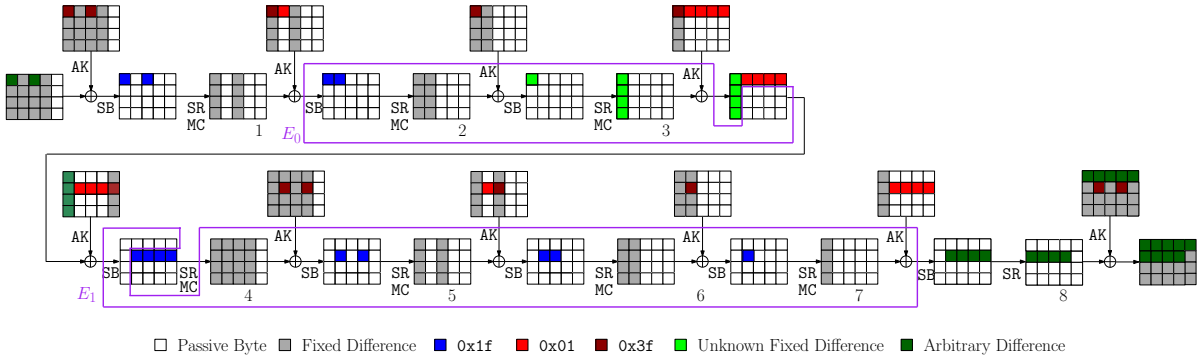


Fig. 7. The Related-key Rectangle Attack on 8-Round Rijndael-160/160. Switch is applied in Round 4

Data Collection

1. Collect N structures $G_i = \{U_i, V_i\}$ of 2^{17} plaintexts each, where $1 \le i \le N$, U_i, V_i are the sets of 2^{16} plaintexts of the form

?	c_1	?	c_2	c_3
c_4	c_5	c_6	c_7	c_8
c_9	c_{10}	c_{11}	c_{12}	c_{13}
c_{14}	c_{15}	c_{16}	c_{17}	c_{18}

and

?	$c_1 \oplus$ 3e	?	$c_2 \oplus$ 3e	c_3
$c_4 \oplus$ 1f	$c_5 \oplus$ 1f	$c_6 \oplus$ 1f	$c_7 \oplus$ 1f	c_8
$c_9 \oplus$ 1f	$c_{10} \oplus$ 1f	$c_{11} \oplus$ 1f	$c_{12} \oplus$ 1f	c_{13}
$c_{14} \oplus$ 21	$c_{15} \oplus$ 21	$c_{16} \oplus$ 21	$c_{17} \oplus$ 21	c_{18}

,

respectively, $c_i's$ $(1 \le i \le 18)$ are fixed 8-bit values and '?' takes all possible values.

2. For each structure G_i
   (a) Ask for the encryption of U_i, V_i under K_a and K_b, respectively, to obtain $G_i^1 = \{X_i^1, Y_i^1\}$.
   (b) Ask for the encryption of V_i, U_i under K_a and K_b, respectively, to obtain $G_i^2 = \{X_i^2, Y_i^2\}$.
   (c) Ask for the encryption of U_i, V_i under K_c and K_d, respectively, to obtain $H_i^1 = \{Z_i^1, W_i^1\}$.
   (d) Ask for the encryption of V_i, U_i under K_c and K_d, respectively, to obtain $H_i^2 = \{Z_i^2, W_i^2\}$.
3. Let $T^{1a} = \{X_i^1\}_{1\le i\le N}$, $T^{1b} = \{Y_i^1\}_{1\le i\le N}$, $T^{2a} = \{X_i^2\}_{1\le i\le N}$, $T^{2b} = \{Y_i^2\}_{1\le i\le N}$, $T^{1c} = \{Z_i^1\}_{1\le i\le N}$, $T^{1d} = \{W_i^1\}_{1\le i\le N}$, $T^{2c} = \{Z_i^2\}_{1\le i\le N}$ and $T^{2d} = \{W_i^2\}_{1\le i\le N}$. Then for the case of $(T^{1a}, T^{1b}, T^{1c}, T^{1d})$, we can construct $(2^{16} \cdot 2^{16} \cdot N)^2 = 2^{64} \cdot N^2$ plaintext quartets meeting the input difference of round 1 given in Fig. 7. Similarly we can obtain $2^{64} \cdot N^2$ plaintext quartets satisfying the input difference of round 1 for $(T^{1a}, T^{1b}, T^{2c}, T^{2d})$, $(T^{2a}, T^{2b}, T^{1c}, T^{1d})$ and $(T^{2a}, T^{2b}, T^{2c}, T^{2d})$, respectively, resulting in $2^{64} \cdot N^2 \cdot 4 = 2^{66} \cdot N^2$ plaintext quartets in total. Among these plaintext quartets there are about $2^{66} \cdot N^2 \cdot (2^{-16})^2 \cdot 2^{-251} = 2^{-217} \cdot N^2$ right quartets. Let $2^{-217} \cdot N^2 = 4$, we can deduce that $N = 2^{109.5}$.
4. Next, derive ciphertext quartets (C_a, C_b, C_c, C_d) and corresponding plaintext quartets (P_a, P_b, P_c, P_d) from $(T^{1a}, T^{1b}, T^{1c}, T^{1d})$, $(T^{1a}, T^{1b}, T^{2c}, T^{2d})$, $(T^{2a}, T^{2b}, T^{1c}, T^{1d})$ and $(T^{2a}, T^{2b}, T^{2c}, T^{2d})$ in an efficient way, such that (P_a, P_b, P_c, P_d), (C_a, C_b, C_c, C_d) satisfy the input and output differences of rounds 1 and 8, respectively, as shown in Fig. 7. Firstly, for the case of $(T^{1a}, T^{1b}, T^{1c}, T^{1d})$ do the following:
   – Initialize two vectors L^{ac} and L^{bd} consisting of 2^{88} lists L_η^{ac} and L_η^{bd}, respectively, where η corresponds to a 11-byte value (i.e., the bytes (1,4) and (i, j) of a ciphertext, where $2 \le i \le 3$, $0 \le j \le 4$), $L_\eta^{ac} = \{S_\eta^a, S_\eta^c, N_\eta^a, N_\eta^c\}$, $L_\eta^{bd} = \{S_\eta^b, S_\eta^d, N_\eta^b, N_\eta^d\}$, $S_\eta^a, S_\eta^b, S_\eta^c, S_\eta^d$ are the sets of ciphertexts under K_a, K_b, K_c and K_d, respectively, as well as their structure indices, and $N_\eta^a, N_\eta^b, N_\eta^c, N_\eta^d$ denote the cardinalities of the sets $S_\eta^a, S_\eta^b, S_\eta^c$ and S_η^d, respectively.
   – For each ciphertext in T^{1a}, extract the 88-bit value η, then insert the ciphertext and its structure index into the set S_η^a of the corresponding list L_η^{ac} and increase N_η^a by 1. For each ciphertext in T^{1c}, xor it with

00	00	00	00	00
00	00	00	00	3e
1f	1f	1f	1f	1f
1f	1f	1f	1f	1f

and then extract the 88-bit value η. After that, insert the ciphertext and its structure index into the set S_η^c of the corresponding list L_η^{ac} and increase N_η^c by 1. Do similarly for the ciphertexts in T^{1b}, T^{1d} and update the lists L_η^{bd}.

– Keep the lists L_η^{ac} in which both N_η^a and N_η^c are non-zero, and keep the lists L_η^{bd} in which both N_η^b and N_η^d are non-zero. Then derive the ciphertext quartets (C_a, C_b, C_c, C_d) from the remaining lists L_η^{ac} and L_η^{bd} by using following criteria:
  - C_a, C_c are chosen from the same list L_η^{ac}, and C_b, C_d come from the same list $L_{\eta'}^{bd}$.
  - The structure indices of C_a and C_b are the same, and the structure indices of C_c and C_d are the same.

  We obtain around $(2^{16} \cdot 2^{16} \cdot 2^{109.5})^2 \cdot (2^{-88})^2 = 2^{107}$ ciphertext quartets (C_a, C_b, C_c, C_d) and their plaintext quartets (P_a, P_b, P_c, P_d) in this step.

– For each of the 2^{107} quartets (C_a, C_b, C_c, C_d), check whether the following conditions

$$(C_a \oplus C_c)_{0,0} = (C_a \oplus C_c)_{0,1} = \ldots = (C_a \oplus C_c)_{0,4}$$

and

$$(C_b \oplus C_d)_{0,0} = (C_b \oplus C_d)_{0,1} = \ldots = (C_b \oplus C_d)_{0,4}$$

hold or not. If not, discard the quartet. The expected number of remaining quartets after this step is about $2^{107} \cdot (2^{-32})^2 = 2^{43}$.

Do similarly for $(T^{1a}, T^{1b}, T^{2c}, T^{2d})$, $(T^{2a}, T^{2b}, T^{1c}, T^{1d})$ and $(T^{2a}, T^{2b}, T^{2c}, T^{2d})$, respectively, and finally we get about $2^{43} \cdot 4 = 2^{45}$ ciphertext quartets (C_a, C_b, C_c, C_d) and their plaintext quartets (P_a, P_b, P_c, P_d).

Key Recovery

5. Guess the subkey bytes $(K_a)_{0,j}$, $(K_c)_{0,j}$ $(j \in \{0, 2\})$ as follows:
   (a) guess $(K_a)_{0,0}$, $(K_c)_{0,0}$ and calculate the values of $(K_b)_{0,0}$, $(K_d)_{0,0}$ by using Table 2.
   (b) guess $(K_a)_{0,2}$, $(K_c)_{0,2}$ and derive the values of $(K_b)_{0,2}$, $(K_d)_{0,2}$ from Table 2.
   For each of the remaining quartets in substeps (a)–(b), test whether the corresponding equations

$$SB((P_a)_{0,j} \oplus (K_a)_{0,j}) \oplus SB((P_b)_{0,j} \oplus (K_b)_{0,j}) = \texttt{1f}$$
$$SB((P_c)_{0,j} \oplus (K_c)_{0,j}) \oplus SB((P_d)_{0,j} \oplus (K_d)_{0,j}) = \texttt{1f}$$

   are satisfied or not. If not, discard the quartet. After this step, the number of remaining quartets is about $2^{45} \cdot (2^{-8})^4 = 2^{13}$.

6. Guess the subkey bytes $(K_a^7)_{1,4}$, $(K_b^7)_{1,4}$, $(K_a^8)_{1,j}$, $(K_b^8)_{1,j}$ $(0 \le j \le 3)$ and obtain the values of $(K_c^7)_{1,4}$, $(K_d^7)_{1,4}$, $(K_c^8)_{1,j}$, $(K_d^8)_{1,j}$ according to Table 2. Then for each remaining quartet (C_a, C_b, C_c, C_d), verify whether the following equations

$$SB((K_a^7)_{1,4}) \oplus SB((K_c^7)_{1,4}) \oplus (C_a)_{0,0} \oplus (C_c)_{0,0} = \texttt{21}$$
$$SB((K_b^7)_{1,4}) \oplus SB((K_d^7)_{1,4}) \oplus (C_b)_{0,0} \oplus (C_d)_{0,0} = \texttt{21}$$
$$SB^{-1}((C_a)_{1,j} \oplus (K_a^8)_{1,j}) \oplus SB^{-1}((C_c)_{1,j} \oplus (K_c^8)_{1,j}) = \texttt{01}$$
$$SB^{-1}((C_b)_{1,j} \oplus (K_b^8)_{1,j}) \oplus SB^{-1}((C_d)_{1,j} \oplus (K_d^8)_{1,j}) = \texttt{01}$$

   hold or not. If not, remove the quartet.

7. If the number of the remaining quartets after above steps is two or more, output the corresponding 14 guessed subkey bytes $(K_a)_{0,j_1}$, $(K_c)_{0,j_1}$, $(K_a^7)_{1,4}$, $(K_b^7)_{1,4}$, $(K_a^8)_{1,j_2}$ and $(K_b^8)_{1,j_2}$ $(j_1 \in \{0, 2\},\ 0 \le j_2 \le 3)$ as the correct key information. Otherwise, return to Step 5 and repeat the procedure.

8. If the above 14 subkey bytes are retrieved after Step 7, perform an exhaustive search over all possible values of the remaining 128 bits of K_a^8 so as to recover the secret key.

5.1 Analysis of the attack

In Step 6, ten equations (each with probability 2^{-8}) need to be satisfied. Therefore, for a wrong guess of the above 14 subkey bytes, the expected number of quartets after Step 6 is $2^{13} \cdot (2^{-8})^{10} = 2^{-67}$. On the other hand, for a right guess of the key, the expected number of right quartets is about 4. This means that we can discard all the wrong subkeys (since the expected number of remaining quartets for a wrong subkey is 2^{-67}) and find the right 14 subkey bytes.

The probability of outputting a wrong key guess in Step 7 is derived by the following Poisson distribution:

$$X \ \sim \ Poi(\lambda = 2^{-67}).$$

As $\Pr[X \geq 2] \approx 2^{-135}$, the expected number of wrong key guesses suggested in Step 7 is about $(2^8)^{14} \cdot 2^{-135} = 2^{-23}$, and the wrong key information can be easily removed in Step 8. Similarly, the probability that two or more quartets remain after Step 7 for the correct key guess is also computed by the Poisson distribution:

$$X \ \sim \ Poi(\lambda = 4).$$

Since $\Pr[X \geq 2] \approx 0.91$, the success probability of the attack on 8-round Rijndael-160/160 is approximately 91%.

5.2 Complexity Issues

The data complexity of this attack is $2^{109.5} \cdot 2^{17} = 2^{126.5}$ chosen plaintexts which are encrypted under K_a, K_b, K_c and K_d, respectively (resulting in $2^{126.5} \cdot 4 = 2^{128.5}$ ciphertexts). The memory complexity is primarily owing to keeping T^{1a}, T^{1b}, T^{1c}, T^{1d}, T^{2a}, T^{2b}, T^{2c} and T^{2d}, thus it can be estimated as $8 \cdot 2^{125.5} \cdot 20 \approx 2^{132.82}$ bytes.

The time complexity of the attack can be derived as follows:

- For the *data collection* phase, the time complexity comes from Step 2 and Step 4.
  - The time complexity of Step 2 is $2^{126.5} \cdot 4 = 2^{128.5}$ 8-round Rijndael-160/160 encryptions.
  - The time complexity of Step 4 can be estimated as $2^{125.5} \cdot 8 = 2^{128.5}$ memory accesses, which can be measured as $2^{128.5} \cdot \frac{1}{20 \cdot 8} \approx 2^{121.18}$ 8-round Rijndael-160/160 encryptions.
- For the *key recovery* phase, the time complexity is calculated as follows:
  - The time complexity of Step 5 can be estimated as $2^{45} \cdot 2^{16} \cdot \frac{4}{20 \cdot 8} + 2^{29} \cdot 2^{32} \cdot \frac{4}{20 \cdot 8} \approx 2^{56.68}$ 8-round Rijndael-160/160 encryptions.
  - The time complexity of Step 6 can be estimated as $2^{13} \cdot 2^{112} \cdot \frac{20}{20 \cdot 8} = 2^{122}$ 8-round Rijndael-160/160 encryptions.
  - The time complexity of Step 8 is about 2^{128} 8-round Rijndael-160/160 encryptions.

As a result, the total time complexity of the attack is approximately $2^{129.28}$ 8-round Rijndael-160/160 encryptions.

6 Attack on 10-Round Rijndael-192/192

By using the 8-round distinguisher for round 2–9 given in Subsection 4, we now present a key recovery attack on 10-round Rijndael-192/192 (round 1–10). Based on Fig. 8, an adversary needs to collect sufficient plaintext quartets such that among these plaintext quartets there are averagely 8 right quartets with respect to the 8-round distinguisher. Then among all the collected plaintext quartets, the expected number of quartets satisfying the input and output differences of this distinguisher for a random permutation is $(8/2^{-355}) \cdot 2^{-384} = 2^{-26}$. From this the adversary could distinguish the correct value from the wrong guessed values of the subkey bytes adopted in rounds 1 and 10 which are related to the 8-round distinguisher with a high probability. The attack procedure is divided into two phases: *data collection* phase and *key recovery* phase.

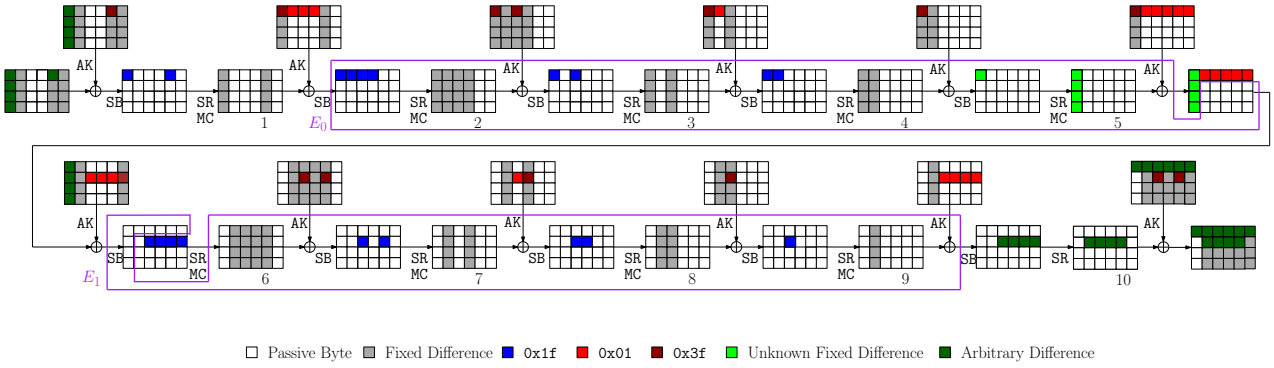


Fig. 8. The Related-key Rectangle Attack on 10-Round Rijndael-192/192. Switch is applied in Round 6

Data Collection

1. Collect N structures $G_i = \{U_i, V_i\}$ of 2^{41} plaintexts each, where $1 \le i \le N$, U_i, V_i are the sets of 2^{40} plaintexts of the form

?	c_1	c_2	c_3	?	c_4
?	c_5	c_6	c_7	c_8	c_9
?	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}
?	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}

and

?	$c_1 \oplus$ 3e	c_2	c_3	?	$c_4 \oplus$ 3e
?	$c_5 \oplus$ 1f	c_6	c_7	$c_8 \oplus$ 1f	$c_9 \oplus$ 1f
?	$c_{10} \oplus$ 1f	c_{11}	c_{12}	$c_{13} \oplus$ 1f	$c_{14} \oplus$ 1f
?	$c_{15} \oplus$ 21	c_{16}	c_{17}	$c_{18} \oplus$ 21	$c_{19} \oplus$ 21

   respectively, $c_i's$ $(1 \le i \le 16)$ are fixed 8-bit values and '?' takes all possible values.

2. For each structure G_i
   (a) Ask for the encryption of U_i, V_i under K_a and K_b, respectively, to obtain $G_i^1 = \{X_i^1, Y_i^1\}$.
   (b) Ask for the encryption of V_i, U_i under K_a and K_b, respectively, to obtain $G_i^2 = \{X_i^2, Y_i^2\}$.
   (c) Ask for the encryption of U_i, V_i under K_c and K_d, respectively, to obtain $H_i^1 = \{Z_i^1, W_i^1\}$.
   (d) Ask for the encryption of V_i, U_i under K_c and K_d, respectively, to obtain $H_i^2 = \{Z_i^2, W_i^2\}$.

3. Let $T^{1a} = \{X_i^1\}_{1\le i\le N}$, $T^{1b} = \{Y_i^1\}_{1\le i\le N}$, $T^{2a} = \{X_i^2\}_{1\le i\le N}$, $T^{2b} = \{Y_i^2\}_{1\le i\le N}$, $T^{1c} = \{Z_i^1\}_{1\le i\le N}$, $T^{1d} = \{W_i^1\}_{1\le i\le N}$, $T^{2c} = \{Z_i^2\}_{1\le i\le N}$ and $T^{2d} = \{W_i^2\}_{1\le i\le N}$. Then for the case of $(T^{1a}, T^{1b}, T^{1c}, T^{1d})$, we can construct $(2^{40} \cdot 2^{40} \cdot N)^2 = 2^{160} \cdot N^2$ plaintext quartets meeting the input difference of round 1 given in Fig. 8. Similarly we can obtain $2^{160} \cdot N^2$ plaintext quartets satisfying the input difference of round 1 for $(T^{1a}, T^{1b}, T^{2c}, T^{2d})$, $(T^{2a}, T^{2b}, T^{1c}, T^{1d})$ and $(T^{2a}, T^{2b}, T^{2c}, T^{2d})$, respectively, resulting in $2^{160} \cdot N^2 \cdot 4 = 2^{162} \cdot N^2$ plaintext quartets in total. Among these plaintext quartets there are about $2^{162} \cdot N^2 \cdot (2^{-40})^2 \cdot 2^{-355} = 2^{-273} \cdot N^2$ right quartets. Let $2^{-273} \cdot N^2 = 8$, we can deduce that $N = 2^{138}$.

4. Next, derive ciphertext quartets (C_a, C_b, C_c, C_d) and corresponding plaintext quartets (P_a, P_b, P_c, P_d) from $(T^{1a}, T^{1b}, T^{1c}, T^{1d})$, $(T^{1a}, T^{1b}, T^{2c}, T^{2d})$, $(T^{2a}, T^{2b}, T^{1c}, T^{1d})$ and $(T^{2a}, T^{2b}, T^{2c}, T^{2d})$ in an efficient way, such that (P_a, P_b, P_c, P_d), (C_a, C_b, C_c, C_d) satisfy the input and output differences of rounds 1 and 10, respectively, as shown in Fig. 8. Firstly, for the case of $(T^{1a}, T^{1b}, T^{1c}, T^{1d})$ do the following:
   – Initialize two vectors L^{ac} and L^{bd} consisting of 2^{112} lists L_η^{ac} and L_η^{bd}, respectively, where η corresponds to a 14-byte value (i.e., the bytes (1,0), (1,5) and (i, j) of a ciphertext, where $2 \le i \le 3$, $0 \le j \le 5$), $L_\eta^{ac} = \{S_\eta^a, S_\eta^c, N_\eta^a, N_\eta^c\}$, $L_\eta^{bd} = \{S_\eta^b, S_\eta^d, N_\eta^b, N_\eta^d\}$, $S_\eta^a, S_\eta^b, S_\eta^c, S_\eta^d$ are the sets of ciphertexts under K_a, K_b, K_c and K_d, respectively, as well as their structure indices, and $N_\eta^a, N_\eta^b, N_\eta^c, N_\eta^d$ denote the cardinalities of the sets $S_\eta^a, S_\eta^b, S_\eta^c$ and S_η^d, respectively.
   – For each ciphertext in T^{1a}, extract the 112-bit value η, then insert the ciphertext and its structure index into the set S_η^a of the corresponding list L_η^{ac} and increase N_η^a by 1. For each ciphertext in T^{1c}, xor it with

00	00	00	00	00	00
00	00	00	00	00	3e
00	1f	1f	1f	1f	1f
00	1f	1f	1f	1f	1f

and then extract the 112-bit value η. After that, insert the ciphertext and its structure index into the set S^c_η of the corresponding list L^{ac}_η and increase N^c_η by 1. Do similarly for the ciphertexts in T^{1b}, T^{1d} and update the lists L^{bd}_η.

- Discard the lists L^{ac}_η in which N^a_η or N^c_η is 0, and remove the lists L^{bd}_η in which N^b_η or N^d_η is 0. For each remaining list L^{ac}_η, initialize 2^{48} lists $L^{ac}_{\eta,\theta} = \{S^a_{\eta,\theta}, S^c_{\eta,\theta}, N^a_{\eta,\theta}, N^c_{\eta,\theta}\}$ defined similarly to L^{ac}_η, where θ corresponds to a 6-byte value (i.e., the bytes $(0, j)$, $0 \le j \le 5$), then do the following:
  - For each ciphertext in S^a_η, extract the 48-bit value θ, then insert the ciphertext and its structure index into the set $S^a_{\eta,\theta}$ of the corresponding list $L^{ac}_{\eta,\theta}$ and increase $N^a_{\eta,\theta}$ by 1.
  - Let $\delta_1, \delta_2, \ldots, \delta_{127}$ denote all possible output differences of the S-Box for the input difference 01. For each ciphertext in S^c_η, xor it with

00	21	21	21	21	21
00	00	00	00	00	00
00	00	00	00	00	00
00	00	00	00	00	00

    and then extract the 48-bit value θ. After that, insert the ciphertext and its structure index into the set S^c_{η,θ_1}, S^c_{η,θ_2}, …, $S^c_{\eta,\theta_{127}}$ of the lists L^{ac}_{η,θ_1}, L^{ac}_{η,θ_2}, …, $L^{ac}_{\eta,\theta_{127}}$ and increase N^c_{η,θ_1}, N^c_{η,θ_2}, …, $N^c_{\eta,\theta_{127}}$ by 1, respectively, where θ_1, …, θ_{127} denote $\theta \oplus (\delta_1\|\delta_1\|\delta_1\|\delta_1\|\delta_1\|\delta_1)$, …, $\theta \oplus (\delta_{127}\|\delta_{127}\|\delta_{127}\|\delta_{127}\|\delta_{127}\|\delta_{127})$, respectively.

  For each remaining list L^{bd}_η, do similarly to get the lists $L^{bd}_{\eta,\theta}$.

- Keep the lists $L^{ac}_{\eta,\theta}$ in which both $N^a_{\eta,\theta}$ and $N^c_{\eta,\theta}$ are non-zero, and keep the lists $L^{bd}_{\eta,\theta}$ in which both $N^b_{\eta,\theta}$ and $N^d_{\eta,\theta}$ are non-zero. Then derive the ciphertext quartets (C_a, C_b, C_c, C_d) from the remaining lists $L^{ac}_{\eta,\theta}$ and $L^{bd}_{\eta,\theta}$ by using following criteria:
  - C_a, C_c are chosen from the same list $L^{ac}_{\eta,\theta}$, and C_b, C_d come from the same list $L^{bd}_{\eta',\theta'}$.
  - The structure indices of C_a and C_b are the same, and the structure indices of C_c and C_d are the same.

With the above procedure we obtain around $(2^{40} \cdot 2^{40} \cdot 2^{138})^2 \cdot (2^{-112})^2 \cdot (\frac{127}{2^{48}})^2 \approx 2^{130}$ ciphertext quartets (C_a, C_b, C_c, C_d) and their plaintext quartets (P_a, P_b, P_c, P_d). Do similarly for $(T^{1a}, T^{1b}, T^{2c}, T^{2d})$, $(T^{2a}, T^{2b}, T^{1c}, T^{1d})$ and $(T^{2a}, T^{2b}, T^{2c}, T^{2d})$, respectively, and finally we get about $2^{130} \cdot 4 = 2^{132}$ ciphertext quartets (C_a, C_b, C_c, C_d) and their plaintext quartets (P_a, P_b, P_c, P_d).

Key Recovery

5. Guess the subkey bytes $(K_a)_{j,5}, (K_c)_{j,5}$ $(j \in \{2, 3, 0\})$ as follows:
   (a) Guess the subkey bytes $(K_a)_{2,5}$, $(K_c)_{2,5}$. According to the key schedule and Table 3, derive $(\Delta K_{ab})_{1,0}$, $(\Delta K_{cd})_{1,0}$ as below:

$$(\Delta K_{ab})_{1,0} = SB((K_a)_{2,5}) \oplus SB((K_a)_{2,5} \oplus 1\text{f}) \oplus 1\text{f},$$

$$(\Delta K_{cd})_{1,0} = SB((K_c)_{2,5}) \oplus SB((K_c)_{2,5} \oplus 1\text{f}) \oplus 1\text{f}.$$

   (b) Guess the subkey bytes $(K_a)_{3,5}$, $(K_c)_{3,5}$. Similarly, compute $(\Delta K_{ab})_{2,0}, (\Delta K_{cd})_{2,0}$ in terms of the key schedule and Table 3.
   (c) Guess the subkey bytes $(K_a)_{0,5}$, $(K_c)_{0,5}$. Calculate $(\Delta K_{ab})_{3,0}, (\Delta K_{cd})_{3,0}$ from the key schedule and Table 3.

   For each of the remaining quartets in substeps (a)–(c), test whether the corresponding equations

$$(P_a)_{((j-1) \bmod 4),0} \oplus (P_b)_{((j-1) \bmod 4),0} \oplus (\Delta K_{ab})_{((j-1) \bmod 4),0} = 0$$
$$(P_c)_{((j-1) \bmod 4),0} \oplus (P_d)_{((j-1) \bmod 4),0} \oplus (\Delta K_{cd})_{((j-1) \bmod 4),0} = 0$$

   are satisfied or not. If not, discard the quartet. After this step, the number of remaining quartets is about $2^{132} \cdot (2^{-8})^6 = 2^{84}$.

6. Guess the subkey bytes $(K_a)_{0,4}$, $(K_c)_{0,4}$. Then for each remaining quartet (P_a, P_b, P_c, P_d), check whether the equations

$$SB((P_a)_{0,4} \oplus (K_a)_{0,4}) \oplus SB((P_b)_{0,4} \oplus (K_a)_{0,4} \oplus 3\mathrm{f}) = 1\mathrm{f}$$
$$SB((P_c)_{0,4} \oplus (K_c)_{0,4}) \oplus SB((P_d)_{0,4} \oplus (K_c)_{0,4} \oplus 3\mathrm{f}) = 1\mathrm{f}$$

hold or not. If not, remove the quartet. The expected number of remaining quartets after this step is $2^{84} \cdot (2^{-8})^2 = 2^{68}$.

7. Guess the subkey bytes $(K_a)_{0,0}$, $(K_b)_{0,0}$, $(K_c)_{0,0}$, $(K_d)_{0,0}$. Then for each remaining quartet (P_a, P_b, P_c, P_d), test whether the equations

$$SB((P_a)_{0,0} \oplus (K_a)_{0,0}) \oplus SB((P_b)_{0,0} \oplus (K_b)_{0,0}) = 1\mathrm{f}$$
$$SB((P_c)_{0,0} \oplus (K_c)_{0,0}) \oplus SB((P_d)_{0,0} \oplus (K_d)_{0,0}) = 1\mathrm{f}$$

hold or not. If not, remove the quartet. The expected number of remaining quartets after this step is $2^{68} \cdot (2^{-8})^2 = 2^{52}$.

8. Guess the subkey bytes $(K_a^9)_{1,5}$, $(K_b^9)_{1,5}$. According to the key schedule and Table 3, derive $(\Delta K_{ac}^{10})_{0,0}$, $(\Delta K_{bd}^{10})_{0,0}$ as below:

$$(\Delta K_{ac}^{10})_{0,0} = SB((K_a^9)_{1,5}) \oplus SB((K_a^9)_{1,5} \oplus 01),$$

$$(\Delta K_{bd}^{10})_{0,0} = SB((K_b^9)_{1,5}) \oplus SB((K_b^9)_{1,5} \oplus 01).$$

Then for each remaining quartet (C_a, C_b, C_c, C_d), verify whether the equations

$$(C_a)_{0,0} \oplus (C_c)_{0,0} \oplus (\Delta K_{ac}^{10})_{0,0} = 0$$
$$(C_b)_{0,0} \oplus (C_d)_{0,0} \oplus (\Delta K_{bd}^{10})_{0,0} = 0$$

hold or not. If not, remove the quartet. Note that $(\Delta K_{ac}^{10})_{0,0}$, $(\Delta K_{bd}^{10})_{0,0} \in \{\delta_1, \ldots, \delta_{127}\}$, and from Step 4 we know that $(C_a)_{0,0} \oplus (C_c)_{0,0}$, $(C_b)_{0,0} \oplus (C_d)_{0,0} \in \{\delta_1, \ldots, \delta_{127}\}$, thus the expected number of remaining quartets after this step is $2^{52} \cdot (2^{-7})^2 = 2^{38}$. Moreover, according to Step 4 and Table 3 we have that for the remaining quartets, $(C_a)_{0,i} \oplus (C_c)_{0,i} = (\Delta K_{ac}^{10})_{0,i}$ and $(C_b)_{0,i} \oplus (C_d)_{0,i} = (\Delta K_{bd}^{10})_{0,i}$ hold for $1 \le i \le 5$.

9. Guess the subkey bytes $(K_a^{10})_{1,j}$, $(K_b^{10})_{1,j}$ $(1 \le j \le 4)$ as follows:
   (a) guess $(K_a^{10})_{1,1}$, $(K_b^{10})_{1,1}$ and calculate the values of $(K_c^{10})_{1,1}$, $(K_d^{10})_{1,1}$ by using Table 3.
   (b) guess $(K_a^{10})_{1,2}$, $(K_b^{10})_{1,2}$ and derive the values of $(K_c^{10})_{1,2}$, $(K_d^{10})_{1,2}$ from Table 3.
   (c) guess $(K_a^{10})_{1,3}$, $(K_b^{10})_{1,3}$, $(K_a^{10})_{1,4}$, $(K_b^{10})_{1,4}$ and use Table 3 to obtain the values of $(K_c^{10})_{1,3}$, $(K_d^{10})_{1,3}$, $(K_c^{10})_{1,4}$, $(K_d^{10})_{1,4}$.

   For each of the remaining quartets in substeps (a)–(c), check whether the corresponding equations

$$SB^{-1}((C_a)_{1,j} \oplus (K_a^{10})_{1,j}) \oplus SB^{-1}((C_c)_{1,j} \oplus (K_c^{10})_{1,j}) = \mathtt{01}$$
$$SB^{-1}((C_b)_{1,j} \oplus (K_b^{10})_{1,j}) \oplus SB^{-1}((C_d)_{1,j} \oplus (K_d^{10})_{1,j}) = \mathtt{01}$$

   are satisfied or not. If not, discard the quartet.

10. If the number of the remaining quartets after above steps is six or more, output the corresponding 22 guessed subkey bytes $(K_a)_{i,5}$, $(K_c)_{i,5}$, $(K_a)_{0,4}$, $(K_c)_{0,4}$, $(K_a)_{0,0}$, $(K_b)_{0,0}$, $(K_c)_{0,0}$, $(K_d)_{0,0}$, $(K_a^9)_{1,5}$, $(K_b^9)_{1,5}$, $(K_a^{10})_{1,j}$ and $(K_b^{10})_{1,j}$ $(i \in \{0, 2, 3\},\ 1 \le j \le 4)$ as the correct key information. Otherwise, return to Step 5 and repeat the procedure.

11. If the above 22 subkey bytes are retrieved after Step 10, perform an exhaustive search over all possible values of the remaining 152 bits of K_a so as to recover the secret key.

6.1 Analysis of the attack

The expected number of remaining quartets after each substep in Step 9 is given as (a) $2^{38} \cdot (2^{-8})^2 = 2^{22}$, (b) $2^{22} \cdot (2^{-8})^2 = 2^6$ and (c) $2^6 \cdot (2^{-8})^4 = 2^{-26}$, respectively. Thus for a wrong guess of the above 22 subkey bytes, the expected number of quartets after Step 9 is 2^{-26}. On the other hand, for a right guess of the key, the expected number of right quartets is 8.

The probability of outputting a wrong key guess in Step 10 is derived by the following Poisson distribution:

$$X \ \sim \ Poi(\lambda = 2^{-26}).$$

As $\Pr[X \geq 6] \approx 2^{-165.49}$, the expected number of wrong key guesses suggested in Step 10 is about $(2^8)^{22} \cdot 2^{-165.49} = 2^{10.51}$, and the wrong key information can be removed in Step 11. Similarly, the probability that six or more quartets remain after Step 9 for the correct key guess is also computed by the Poisson distribution:

$$X \ \sim \ Poi(\lambda = 8).$$

Since $\Pr[X \geq 6] \approx 0.81$, the success probability of the attack on 10-round Rijndael-192/192is approximately 81%.

6.2 Complexity Issues

The data complexity of this attack is $2^{138} \cdot 2^{41} = 2^{179}$ chosen plaintexts which are encrypted under K_a, K_b, K_c and K_d, respectively (resulting in $2^{179} \cdot 4 = 2^{181}$ ciphertexts). The memory complexity is primarily owing to keeping T^{1a}, T^{1b}, T^{1c}, T^{1d}, T^{2a}, T^{2b}, T^{2c} and T^{2d}, thus it can be estimated as $8 \cdot 2^{178} \cdot 24 \approx 2^{185.59}$ bytes.

The time complexity of the attack can be derived as follows:

- For the *data collection* phase, the time complexity comes from Step 2 and Step 4.
  - The time complexity of Step 2 is $2^{179} \cdot 4 = 2^{181}$ 10-round Rijndael-192/192encryptions.
  - The time complexity of Step 4 can be estimated as $2^{178} \cdot 8 = 2^{181}$ memory accesses, which can be measured as $2^{181} \cdot \frac{1}{24 \cdot 10} \approx 2^{173.09}$ 10-round Rijndael-192/192encryptions.
- For the *key recovery* phase, the time complexity is calculated as follows:
  - The time complexity of Step 5 can be estimated as $2^{132} \cdot 2^{16} \cdot \frac{4}{24 \cdot 10} + 2^{116} \cdot 2^{32} \cdot \frac{4}{24 \cdot 10} + 2^{100} \cdot 2^{48} \cdot \frac{4}{24 \cdot 10} \approx 2^{143.68}$ 10-round Rijndael-192/192encryptions.
  - The time complexity of Step 6 can be estimated as $2^{84} \cdot 2^{64} \cdot \frac{4}{24 \cdot 10} = 2^{142.09}$ 10-round Rijndael-192/192encryptions.
  - The time complexity of Step 7 can be estimated as $2^{68} \cdot 2^{96} \cdot \frac{4}{24 \cdot 10} = 2^{158.09}$ 10-round Rijndael-192/192encryptions.
  - The time complexity of Step 8 can be estimated as $2^{52} \cdot 2^{112} \cdot \frac{4}{24 \cdot 10} = 2^{158.09}$ 10-round Rijndael-192/192encryptions.
  - The time complexity of Step 9 can be estimated as $2^{38} \cdot 2^{128} \cdot \frac{4}{24 \cdot 10} + 2^{22} \cdot 2^{144} \cdot \frac{4}{24 \cdot 10} + 2^6 \cdot 2^{176} \cdot \frac{8}{24 \cdot 10} \approx 2^{177.09}$ 10-round Rijndael-192/192encryptions.
  - The time complexity of Step 11 is about $2^{152} \cdot 2^{10.51} = 2^{162.51}$ 10-round Rijndael-192/192encryptions.

As a result, the total time complexity of the attack is approximately $2^{181.09}$ 10-round Rijndael-192/192encryptions.

7 Conclusion

In this paper, we performed key recovery attacks on reduced-round versions of Rijndael-160/160/160 and Rijndael-192/192. Firstly, we constructed a 6-round related-key rectangle distinguisher of Rijndael-160/160/160, with which we attacked 8 rounds of the cipher. Moreover, we established a 8-round related-key rectangle distinguisher of Rijndael-192/192, based on which we demonstrated the attack on 10 rounds of the cipher. Our results show that the related-key rectangle attack is one of the best methods to analyze Rijndael and Rijndael-like structures. To sum up, none of our attacks directly threatens the security of Rijndael but they reduce the security margin of the cipher.

References

1. National Bureau of Standards. Data Encryption Standard. FIPS-Pub.46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.
2. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002.
3. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES*, pages 450–466, 2007.
4. FIPS 197. Advanced Encryption Standard. Federal Information Processing Standards Publication 197, U.S. Department of Commerce/N.I.S.T , 2001.
5. Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The Block Cipher Square. In Eli Biham, editor, *FSE*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 1997.
6. Henri Gilbert and Marine Minier. A Collision Attack on 7 Rounds of Rijndael. In *AES Candidate Conference*, pages 230–241, 2000.
7. Stefan Lucks. Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys. In *AES Candidate Conference*, pages 215–229, 2000.
8. Alex Biryukov. The Boomerang Attack on 5 and 6-Round Reduced AES. In Hans Dobbertin, Vincent Rijmen, and Aleksandra Sowa, editors, *AES Conference*, volume 3373 of *Lecture Notes in Computer Science*, pages 11–15. Springer, 2004.
9. Raphael Chung-Wei Phan. Impossible Differential Cryptanalysis of 7-round Advanced Encryption Standard (AES). *Inf. Process. Lett.*, 91(1):33–38, 2004.
10. Wentao Zhang, Wenling Wu, and Dengguo Feng. New Results on Impossible Differential Cryptanalysis of Reduced AES. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *ICISC*, volume 4817 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2007.
11. Hüseyin Demirci and Ali Aydin Selçuk. A Meet-in-the-Middle Attack on 8-Round AES. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 116–126. Springer, 2008.
12. Jiqiang Lu, Orr Dunkelman, Nathan Keller, and Jongsung Kim. New Impossible Differential Attacks on AES. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pages 279–293. Springer, 2008.
13. Eli Biham, Orr Dunkelman, and Nathan Keller. Related-Key Impossible Differential Attacks on 8-Round AES-192. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 21–33. Springer, 2006.
14. Wentao Zhang, Wenling Wu, Lei Zhang, and Dengguo Feng. Improved Related-Key Impossible Differential Attacks on Reduced-Round AES-192. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 15–27. Springer, 2006.
15. Jongsung Kim, Seokhie Hong, and Bart Preneel. Related-Key Rectangle Attacks on Reduced AES-192 and AES-256. In Alex Biryukov, editor, *FSE*, volume 4593 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2007.
16. Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-Key Differential-Linear Attacks on Reduced AES-192. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *INDOCRYPT*, volume 4859 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 2007.
17. Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and Related-Key Attack on the Full AES-256. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer, 2009.
18. Alex Biryukov and Dmitry Khovratovich. Related-Key Cryptanalysis of the Full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.
19. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David Wagner, and Doug Whiting. Improved Cryptanalysis of Rijndael. In Schneier [46], pages 213–230.
20. Jorge Nakahara, Daniel Santana de Freitas, and Raphael Chung-Wei Phan. New Multiset Attacks on Rijndael with Large Blocks. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt*, volume 3715 of *Lecture Notes in Computer Science*, pages 277–295. Springer, 2005.
21. Samuel Galice and Marine Minier. Improving Integral Attacks Against Rijndael-256 Up to 9 Rounds. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2008.
22. Yanjun Li and Wenling Wu. Improved Integral Attacks on Rijndael. *Journal of Information Science and Engineering*, 27(6):2031–2045, 2011.

23. Jorge Nakahara and Ivan Carlos Pavão. Impossible-Differential Attacks on Large-Block Rijndael. In Juan A. Garay, Arjen K. Lenstra, Masahiro Mambo, and René Peralta, editors, *ISC*, volume 4779 of *Lecture Notes in Computer Science*, pages 104–117. Springer, 2007.
24. Lei Zhang, Wenling Wu, Je Hong Park, Bonwook Koo, and Yongjin Yeom. Improved Impossible Differential Attacks on Large-Block Rijndael. In Tzong-Chen Wu, Chin-Laung Lei, Vincent Rijmen, and Der-Tsai Lee, editors, *ISC*, volume 5222 of *Lecture Notes in Computer Science*, pages 298–315. Springer, 2008.
25. Qingju Wang, Dawu Gu, Vincent Rijmen, Ya Liu, Jiazhe Chen, and Andrey Bogdanov. Improved Impossible Differential Attacks on Large-Block Rijndael. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *ICISC*, volume 7839 of *Lecture Notes in Computer Science*, pages 126–140. Springer, 2012.
26. Paulo S. L. M. Barreto, Ventzislav Nikov, Svetla Nikova, Vincent Rijmen, and Elmar Tischhauser. Whirlwind: A New Cryptographic Hash Function. *Des. Codes Cryptography*, 56(2-3):141–162, 2010.
27. Eli Biham and Orr Dunkelman. The SHAvite-3 Hash Function. Submission to NIST (Round 2), 2009.
28. P.S.L.M. Barreto and V. Rijmen. The whirlpool hashing function. Submitted to NESSIE (September 2000), 2000. `http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html`(2008/12/11) (revised May 2003).
29. R. Benadjila, O. Billet, H. Gilbert, G. Macario-Rat, T. Peyrin, M. Robshaw, and Y. Seurin. SHA-3 proposal: ECHO. Submission to NIST (updated), 2009. `http://crypto.rd.francetelecom.com/echo/doc/echo_description_1-5.pdf`.
30. Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON Family of Lightweight Hash Functions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.
31. Praveen Gauravaram, Lars R. Knudsen, K. Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Grøstl - a SHA-3 candidate. Submission to NIST, 2008. `http://www.groestl.info`.
32. Eli Biham, Orr Dunkelman, and Nathan Keller. The Rectangle Attack - Rectangling the Serpent. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 340–357. Springer, 2001.
33. Eli Biham. New types of cryptanalytic attacks using related keys. *J. Cryptology*, 7(4):229–246, 1994.
34. John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *Information and Communication Security, First International Conference, ICICS'97, Beijing, China, November 11-14, 1997, Proceedings*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 1997.
35. Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit WEP in less than 60 seconds. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *Information Security Applications, 8th International Workshop, WISA 2007, Jeju Island, Korea, August 27-29, 2007, Revised Selected Papers*, volume 4867 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 2007.
36. Eli Biham, Orr Dunkelman, and Nathan Keller. Related-Key Boomerang and Rectangle Attacks. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 507–525. Springer, 2005.
37. Alex Biryukov and Ivica Nikolic. Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer, 2010.
38. Eli Biham, Orr Dunkelman, and Nathan Keller. A Related-Key Rectangle Attack on the Full KASUMI. In Bimal K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2005.
39. Orr Dunkelman, Nathan Keller, and Adi Shamir. A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2010.
40. David Wagner. The Boomerang Attack. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *LNCS*, pages 156–170. Springer, 1999.
41. Marine Minier and Benjamin Pousse. Improving Integral Cryptanalysis against Rijndael with Large Blocks. *CoRR*, abs/0910.2153, 2009.
42. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In Schneier [46], pages 75–93.
43. Eli Biham, Orr Dunkelman, and Nathan Keller. New Combined Attacks on Block Ciphers. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *LNCS*, pages 126–144. Springer, 2005.
44. Florent Chabaud and Antoine Joux. Differential Collisions in SHA-0. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 56–71. Springer, 1998.
45. Alex Biryukov, Christophe De Cannière, and Gustaf Dellkrantz. Cryptanalysis of SAFER++. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2003.
46. Bruce Schneier, editor. *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*. Springer, 2001.